

REMARKS

Summary of Examiner's Action

In the subject office action, the Examiner rejected claims 1-12 and 23-26 under 35 USC 102(b) as being fully anticipated by Christie et al (USP 6,154,818), and claims 13, 15, 16, 19-22 by Parmar (USP 3,916,385).

Claims 17 and 18 were objected to, but allowable if rewritten in independent form including the limitations of the base and intervening claims.

102(b) rejection based on Christie ('818)

Claim 1 recites

A processor comprising:

a control register to store a task privilege level for a task; and

*a **privilege remapper** coupled to the control register to dynamically remap the stored task privilege level (emphasis added).*

Thus, claim 1, requires a **privilege remapper**. The function of the **privilege remapper** is to **dynamically remap the stored task privilege level**. In other words, the **stored task privilege level** is the "object" or the "operand" of the **remap** operation.

In rejecting claim 1, the Examiner asserts that the remapper (element 306) of Fig. 3b of Christie anticipated the required **privilege remapper** of claim 1. Applicant respectfully disagrees.

In col.14, lines 40-41, Christie clearly states that "remapper 306 is a circuit that maps an MSR address to a local address of an implemented MSR". The input to Christie's MSR is an address, and the output is a local address. Therefore, the function performed by Christie's MSR is an address mapping operation. The

"object" or "operand" of the mapping operation is an address, and has nothing to do with remapping ***stored task privilege level***.

One skilled in the art will appreciate the address mapping operation described is analogous to mapping of virtual address to physical address. Accordingly, the remapper of Christie is an address remapper, not a privilege remapper.

Christie did disclose that access to the address space partition is controlled by a task's privilege. The accessing MSR address is also provided to a validity check circuit 304 to determine if the accessing task has the required privilege before granting access. Validity check circuit 304 also receives the privilege level of the task as input. If the task's privilege level is insufficient to qualify to make the requested access, a fault is generated (see Fig. 3, and its corresponding description in col. 14).

Nothing in Fig. 3 or its corresponding description suggests the address remapper, the validity check circuit, or any other element, individually or in combination, act to ***remap a stored task privilege level***.

Accordingly, claim 1 is patentable over Christie.

Claims 2-6 depend on claim 1, incorporating its limitations. Accordingly, for at least the same reasons, claims 2-6 are also patentable over Christie.

Claims 7, 23 and 25 contain in substance the above discussed limitations of claim 1. Accordingly, for at least the same reasons, claims 7, 23 and 25 are patentable over Christie.

Claims 8-12, 24 and 26 depend on claim 7, 23 and 25 respectively, incorporating their respective limitations. Accordingly, for at least the same reasons, claims 8-12, 24 and 26 are patentable over Christie.

102(b) rejection based on Parmar ('385)

Claim 13 recites in pertinent part

attributing a ring-2 privilege level to a first task, nominally giving said first task more privilege than a second plurality of tasks which are attributed with a ring-3 privilege level; and dynamically remapping each ring-2 privilege level to a ring-3 privilege level, and each ring-3 privilege level to a ring-2 privilege level prior to runtime privilege checking to cause said first task to execute in fact with less privilege.

Thus, claim 13 requires

- an initial assignment of ring privileges to tasks
- moreover, the initial assignment gives a first task a ring-2 privilege, a higher ring privilege than the ring-3 privileges assigned to a second plurality of tasks;
- however, prior to runtime privilege checking (to establish the resources a task may access), the privilege assignment is turned "upside down", with the first task's ring-2 privilege **dynamically remapped** to a ring-3 privilege, and the second plurality of tasks' ring-3 privilege **dynamically remapped** to a ring-2 privilege.

In rejecting claim 13, the Examiner relied on Parmar's teachings in Fig. 2, and the corresponding description in col. 9, line 45-col 11 line 50, in particular, col. 11, lines 20-50. Fig. 2 and the corresponding description merely disclosed the underlying ring mechanism. It did not disclose the required **dynamically remapping of a ring-2 privilege assigned to a task to ring-3, and a ring-3 privilege assigned to a task to ring-2.**

The Examiner relied specifically on Pamar's disclosure in col. 11, lines 20-50, which in pertinent part states

"a procedure in an outer ring such as ring 3 can **branch** to an inner ring such as ring 1 via gate 204 which results in a legal **branch** 203, but a procedure operating in an inner ring such as ring 2 may not **branch** to an outer ring such as ring 3".

One skilled in the art would understand the word **branch** to mean switching execution from one procedure (or task) to another procedure (or task). So the disclosure that "a procedure in an outer ring such as ring 3 can **branch** to an inner ring" merely means that execution may switch from a first procedure (task) with ring 3 privilege to a second procedure (task) with ring 1 privilege. The privileges of the first and second procedures (or tasks) DO NOT change. Therefore, there is no remapping of their privileges.

Even if we are to ignore the meaning of these passages as understood by those ordinarily skilled in the art, and assume arguendo that the word "**branch**" may mean "remapping of the branching procedure's privilege", Parmar still at most merely anticipated the portion of the limitation requiring remapping of ring-3 privilege to ring-2 privilege (low to high). Since Parmar clearly stated that a procedure may not **branch** from ring-2 to ring-3 (high to low) (see col. 11, lines 20-50), Parmar failed to anticipate the remaining limitation requiring not only remapping of ring-3 privilege to ring-2, but a coordinated remapping of ring-2 privilege to ring-3.

Accordingly, Parmar failed to anticipate each and every limitation of claim 13. Therefore, claim 13 is patentable over Parmar.

Claims 16, 19, and 21 contain in substance the above discussed limitations of claim 13, therefore claims 16, 19 and 21 are patentable over Parmar.

Claims 15, 20 and 22 depend on claims 13, 19 and 21 respectively, incorporating their respective limitations. Therefore, for at least the same reason, claims 15, 20 and 22 are patentable over Parmar.

103 rejection based on Parmar ('385)

Claim 14 depends on claim 13 incorporating its limitation.

Since claim 13 is patentable over Parmar, for at least the same reason, claim 14 can not be obvious in view of Parmar. Thus, claim 14 is patentable over Parmar.

Claims 17 and 18

Claims 17 and 18 depend on claim 16. Since claim 16 is patentable over the recited prior art, claim 17 and 18 are allowable without having to be rewritten in independent form.

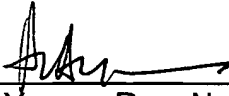
Conclusion

In view of the foregoing, Applicant respectfully submits that claims 1-26 are in condition for allowance, and early issuance of the Notice of Allowance is respectfully requested.

Please charge any shortages and credit any overages to Deposit Account No. 500393.

Respectfully submitted,
Schwabe, Williamson and Wyatt

Date: August 20, 2003



Aloysius AuYeung, Reg. No. 35,432
Attorney for Applicant